Bill Pugsley has over thirty years commercial data processing experience and founded what is now Perwill plc in 1973. He is currently a Board Member of e centre^{UK} (result of merger of the UK ANA and Electronic Commerce Association), Chairman of the e centre^{UK} Vendors Group, executive member of Value Chain Group of e centre^{UK} and editor on the e centre^{UK} Editorial Board.

Bill also represents his Company at **DISA** (North American Standards Organisation), **EEMA** (The European Forum for Electronic Business) and **BASDA** (Business Applications Software Developers Association).

Bill has been involved in EDI - and thus Electronic Commerce, through the US subsidiary of Perwill plc, since 1986 and his Company has developed EC enabling software for EC practitioners that run on a number of Hardware platforms. Perwill is now one of the leading independent providers of Electronic Commerce software and services in the World.

Bill has assisted clients around the world in the introduction of Electronic Commerce and has presented papers and chaired events on Electronic Commerce in five continents.

Bill Pugsley can be reached as follows:

Bill Pugsley
Managing Director
Perwill plc
13a Market Square
Alton
Hampshire
GU34 1UR
ENGLAND
Tel: +44 (0) 1420 545000
Fax: +44 (0) 1420 545001
Email: bill.pugsley@perwill.com

Why is XML so Useful?

The presentation provided by Bill Pugsley gives a perspective to the development of XML and a number of the issues associated with it. It draws from case studies of real usage and goers onto explain the real benefits that can be derived in moving electronic commerce through the complete supply chain.

Why is XML so Useful?

*e**X**tensible **M**arkup **L**anguage (XML) is the ‘lingua franca’ for electronic communication in the current age (2001).*

Historically paper documents have been used to convey information between businesses (Quotations, Orders, Order Acknowledgements, Change Orders, Change Order Acknowledgements through to actual payments).

BASDA (Business Applications Software Developers Association) has analyzed transaction flows in the UK and has assessed that 95% of everything being typed into computers in the United Kingdom (and probably elsewhere in the western hemisphere) is taken from a piece of paper printed by a Computer. Apart from the obvious waste of time that incurs, the problems introduced by re-keying errors are such that the cost is magnified. If 11 (eleven) is typed instead of 1 (one) the costs to the organization responsible may truly be significant.

Over the last twenty years major businesses have been successful in the implementation of electronic trading **(the exchange of business information directly from one computer application to another computer application, using agreed standards, without human intervention)** *with their larger suppliers. Orders are being placed using Electronic Data Interchange (EDI) with delivery of the products ordered in less than ten hours (as demonstrated in the Automotive build market). However, EDI has only been implemented by less than 2% of businesses in the United Kingdom.*

In the past EDI was considered as being the ideal means of delivering data between applications, however it too had its limitations! The implementation of EDI capability was not low cost and thus was unattractive (in investment terms) for low volume, low value transactions.

However, XML with its inherent ability to produce human readable ‘documents’ does not have the restrictions of EDI in terms of cost or usability. As such, it is likely to be accepted by over 60% of businesses (at least 1,000,000 in the United Kingdom) in the same way that facsimile is seen as a mandatory requirement for any business today.

XML unlike its ‘elder brother, EDI’, uses the Internet for its delivery and it facilitates receipt of data in a format that can be read by a human being (EDI messages take some getting used to). It also offers the data in a structured format such that a receiving application may be able to read it without the need to re-key any information.

Background

XML has evolved from Standard Generalized Markup Language (SGML) which was originally conceived for printing and the original language used to build Internet Web Sites Hyper Text Markup Language (HTML).

SGML allowed for the development of a control language that can be used by various printer Manufacturers (and software providers) to ensure consistent reported (printed) output regardless of the origin of the data. HTML provides a general-purpose method of defining screens and their display characteristics for access by a variety of ‘web browsers’.

XML has taken the strengths of both of these facilities and offers a generic design for the delivery of data that can be both displayed and integrated with applications such that data delivered can be read automatically using the facilities offered. With some additional controls it will even permit data to be printed in a structured fashion.

XML has additionally evolved to fulfil the lack of facilities offered by traditional email. Essentially email offers a totally free format for delivery of information that is best suited to receipt by a human being who can then (hope) to interpret the data received (assuming it is presented in a language the reader has knowledge of). By its very free format nature it presents conflicting methods of presentation (and sometimes-conflicting delivery capabilities. It certainly does not readily offer the scope needed to permit data to be delivered in such a way as it can then automatically be 'read' by the receiving application.

So what is XML offering that is different to that offered by email and EDI with their various restrictions?

XML data structures are typically defined using a Tree structure that makes the data easier to manipulate. It contains Tagged information that makes it (potentially) easier to read (by a human being). It provides for Embedded Support and some applications will today support some specific XML standards. As such it will be the delivery method of the future.

By addition of the e**X**tensible **S**tyle sheet **L**anguage (XSL) it is possible to actually simply convert an XML message into a document for simple viewing (or printing).

The major benefit of XML is that the method of defining the potential message (the XML Schema) offers an open (i.e. generic) capability. This can be constrained by the implementation and use of a DTD (Document Type Definition) that defines the full potential data structure (but makes it a "closed" solution).

XML would not have been feasible before 1997 because the processing power of computer systems was not sufficient to deal with the current sizes of data. However, even with the processing power available today, there are still some pitfalls associated with its usage.

Issues Associated with XML

There is the problem of there being over 1000 bodies currently defining "XML standards". Here the "standard" may vary from what the elements are that make up a transaction to how to actually describe the element itself. For example, the value <InvoiceNumber> (when used as a "Tag" does not mean the same as <Inv-No>, <Invoice-Num>, <INVOIC> nor <ABC>. Also, DTD's can change on a whim.

Although a human being can readily 'read' data and make the connection between <Inv-No>, <Invoice-Num>, <INVOIC> or <InvoiceNumber>, a computer system (application) expects to be able to identify a value so that it may be onward processed. An application set up to read <InvoiceNumber> would not be able to recognize <Inv-No>.

Some current initiatives for XML development include BizTalk being promoted by Microsoft and BASDA; RosettaNet - Electronic Components Industry and UN/CEFACT & OASIS Organization for the Advancement of Structured Information Standards.

EDI has historically used third party Value Added Networks (VAN's) that have offered store and forwarding of information and most EDI continues to be carried over these 'proprietary' and often expensive services. There has been a move in recent years to actually use the Internet for the delivery of EDI data but implementations are still in their early stages. XML presupposes that the Internet is actually used (although not exclusively) for the transport of the messages created in that fashion.

Data converted to XML format will suffer from Data Bloating in that data strings will at least double in size. Traditional XML equivalences of the data delivered by EDI may actually be thirty times larger in size. This adds to the cost of transmission of the data (nothing really noteworthy in that as costs for bandwidth have decreased in recent years) but it will require additional processing time as each string of data is read character by character.

Where an EDI (regardless of EDI Standard being used) Invoice may occupy 4,000 characters, its equivalent XML message may occupy as many as 100,000 characters. So an organization sending 10,000 invoices to one trading partner may well wish to (continue to) do so using EDI messaging.

Unlike the traditional EDI, for which there are about twenty-four 'unique' syntax's worldwide, there are over 850 different XML initiatives going on around the globe. Examination of the eBiz XML repository maintained by Microsoft on 31st March 2000 showed that there were 26 different 'flavours' of Invoice Message located on the system. Even assuming that each of the 850 (or more) initiatives has an average of two flavours of a transaction, the result will be a plethora of different messages.

On the face of it the larger company may consider this a nightmare scenario. However, companies like Perwill plc have developed software solutions that minimize the impact of the variety of data structures that companies have to deal with.

Not only does the Perwill solution set offer the ability of reading virtually any format of XML, it can be configured to convert the format received into a variety of any other formats. This would include conversion to traditional EDI formats; other XML formats or in a worst case, print the data in a meaningful fashion such that it can be 'read'. By the same token, products such as Perwill's offer the reverse processing outlined in the foregoing.

Where larger companies have forced their smaller suppliers down the EDI route (prevalent in the Retail sector, especially supermarket supply), the resentment for the 'cost incurred and often lack of tangible benefits has otherwise soured what might have been a good business relationship. This has been exacerbated when certain large retailers have then told their suppliers that the solution originally recommended is no longer promoted and that they need to move to another supplier. This has necessitated additional costs. In fairness, those who did implement EDI trading with their larger customers have found that they retained the trading relationship they had whereas those who refused to implement the technology often found that they were dropped as a supplier.

With the advent of XML it is now possible for a larger company to deliver its business transactions in traditional EDI and XML formats, where XML formats can be targeted to those suppliers who cannot handle EDI.

From a supplier's point of view, Internet access will be required (often nothing more than an email address on a free or low cost service). Usage of a Browser such as Internet Explorer 5 and the associated delivery of XSL (that can be used to 'present XML data in a meaningful way), will allow the recipient of data to be able to view (and if need be) print the received information.

The Internet is now a proven delivery mechanism and delivery of XML data will have the potential to at least match delivery by facsimile (without the need to print – assuming the fax machine has ink and paper). In reality it offers a better means of delivery as multiple copies can be printed as needed.

In itself this would not be a major development save for the fact that BASDA (Business Applications Developers Association) has encouraged its over 300 plus membership (Financial Application product providers whose products range from TAS Books through to SAP R/3) to XML enable their applications. The result of this work has been to create XML templates for sending and receiving Purchase Orders and Invoices (the most critical of transactions being used).

Over 100 BASDA members have indicated that they will implement the eBiz XML messages jointly developed by BASDA and Microsoft (some have already done so). The consequence is that users of those applications will be able to electronically receive (initially) Purchase Orders and be able to send Invoices, without having to re-key any information.

BASDA has been particularly successful in its cooperative working with Microsoft under the BizTalk initiative.

XML and Financial Transactions

Much has been written about using XML to transact all levels of business, from sending a request for quotation to receiving an Invoice. Many people in Europe have the misconception that financial instruments (Invoices, Credit Notes & Debit Notes) that contain Value Added Taxation (VAT) elements are also permitted using XML messaging. However, the reality is that is not so!

In Europe, any company or organisation solely relying on XML Invoices, Credit Notes or Debit notes as being the proof of a transaction who have not agreed the delivery or receipt of such transactions with the relevant VAT authorities for their Countries, otherwise they may be in for an unpleasant surprise at their next Government VAT Audit.

Where single XML financial instruments are sent (typically over the Internet) and a copy is available to both the sender and receiver for the statutory six years (for possible Government audit), there is unlikely to be a problem (but see 28 day notice noted below). However, where several transactions are being sent in the same batch (as a single attachment to an email for example), the sender should also create a summary message that contains hash totals of the value fields of 'each type' of transaction. Thus if Credit Notes are sent with Invoices, there needs to be a summary XML set for Credit Notes and a separate summary set for Invoices. The recipient of the XML message should then recalculate (hash total) the received transactions and compare the calculated results against the received summaries. If there is a mismatch in calculations (which suggest data has been corrupted on route) then either the data can be accepted (and the recipient must tell the sender where the problem lies) or the recipient can reject the batch of data that is in error.

Assuming that the sender has agreed (a mandatory twenty eight days notice is required in the United Kingdom) with their local Government office that they can send XML financial instruments to specific customers, (and the customers have given 28 days notice to their own HMC&E offices) then paper copies do not need to be sent. Without the notice and without the summary reports, paper copies of transactions should be sent and the organisations are expected to declare which is the document for tax purposes (the XML or the paper copy). Without such a declaration and a demonstration that a procedure is in place to use only 'one set of data' for the purposes of reconciling VAT (not two sets of data leading to possible duplication of bookings) it is unlikely that a Company's VAT inspector will accept the delivery of data in duplicate.

The best way forward is to generate the necessary control records (in XML format) and where such data is received to verify the control totals being delivered. Ideally a control print should be produced stating what has been received and what was calculated.

BASDA (Business Application Software Developers Association) in the United Kingdom is working to provide a suggested template for the XML messages needed to provide the ‘controlling data’. The results of their development will be a template that can be followed by businesses wishing to ‘trade’ financial instruments using XML data structures.

PrecisionXML* - An Innovation

It is an extension to XML and a method of dealing with XML extension in such a way that data validation takes place as the XML is being constructed and de-constructed for onward movement within a computer system

Overview

XML has become a popular way of transferring data between computer systems. Because it is often used to transfer ‘one off’ messages between users which are usually viewed and processed by humans, it lacks the control mechanisms that were put in place within other data structures used to transfer data between systems. This lack of control and data validation can make it unsafe to rely on XML for the seamless transfer of data between systems.

Perwill have addressed these substantial issues and have introduced **PrecisionXML** within Perwill ***eBiz-Manager*** to solve these problems.

Perwill licences this technology for use by other technology vendors so that whole communities can benefit from the precision, reliability and Integrity it offers.

Some Definitions

An **XML document** contains **data** to be transferred from A to B. Data is tagged.

A **DTD** describes the **structure** of the XML in a **syntax** specific to DTDs, but can be used to describe SGML as well.

XML-Data is another way of describing an XML document’s **structure** – but uses XML syntax to do it. **XML-Data** is commonly known as an **XML Schema.**

A **Schema** is the generic name given to something that describes the **structure** of a document. Therefore a **Schema** could actually be either a **DTD** or an **XML-Data** file.

A **Schema** is used to validate the **XML document** i.e. the **XML document** is **checked** against the **Schema**.

In order for a **XML document** to be **displayed** as a form it needs to have an Associated **XML Style Sheet** (XSL) and the appropriate Browser installed on the target machine. Currently only Internet Explorer (version 5 and above) has this capability. Netscape have promised it in their next release.

Processing XML

XML process engines should check any XML data received against the relevant Schema as a matter of course.

However the process engine cannot check the type of data unless the engine has special facilities built in to check the data type and data precision. A trivial example from an XML document will suffice to illustrate this. Within the document is a field called **<price>**.

One user sends **<price>$27.00</price>**

Another sends **<price>twenty-seven dollars</price>**

Both are correct and can be interpreted by a human being. However most computer systems would have difficulty deciphering the example given in words rather than digits.

Key Issues For Inter-Operability

All XML data must be properly and unambiguously constructed, leaving no scope for misinterpretation by the target system.

All XML data relating to a transaction should be an integral part of that transaction and not placed after the transaction so the XML process engine has to make assumptions about what is the start of the next transaction.

If the sequence in which the XML documents must be processed is important, then the sequence in which the XML documents left the source environment and the order in which they are to be processed by the target environment must be an integral part of the XML data structure, and not left to the statistical probability that it will be processed in the correct order.

Data must be in the correct format and correctly data-typed for use by the target machine.

Data Validation

It is pointless sending data to another computer system if that system cannot **with 100% certainty** use that data. Therefore data must be correctly tagged, validated against the Schema and correctly data-typed with the right precision and then processed in the correct order. Two examples will suffice to illustrate this – both taken from real life:

The food industry regularly sends data between systems. If a series of XML documents are exchanged between the system used to develop the recipe, and the system which publishes the data about the recipe to the printer responsible for the label production (and therefore the consumer), and they are not correctly sent to the target system and processed in the correct order then there are substantial possibilities for publishing erroneous information which could result not only in the death of an anaphylactic but also – rightly - lay the company open to prosecution.

There is a difference between ‘2.6%’ and ‘2.60%’ when used in a scientific or analytic way as the presence of the trailing zero in the second example shows the test used to process the second sample was ten times more accurate than the test used in the first example.

How does it Work?

In order to make XML safer to use in computer to computer environments, PrecisionXML ensures that an XML document contains all the information necessary so an XML process engine that is compliant with **PrecisionXML** can validate the data before passing it on to the target system – and alert a human being if there is something amiss.

***Perwill has applied for a Trade Mark for PrecisionXML. PrecisionXML** is proprietary technology. It is available for licenced use to other technology vendors. To discuss licencing this technology applies in the first instance to Bill Pugsley at Perwill plc.

So where are the User Benefits?

XML is actually user readable and very easily converted into documents (limited 'configuration' is required rather than full programming or report writing). XML offers an alternative to traditional EDI and as such a more affordable route to those smaller trading organisations.

During the presentation current case studies will be provided to include a review of how XML has been taken up by organisations around the world.